

Real Time Intelligent Target Detection and Analysis with Machine Vision

Ayanna Howard, Curtis Padgett, Kenneth Brown
Machine Vision and Tracking Sensors Group
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109-8099

Abstract

We present an algorithm for detecting a specified set of targets for an Automatic Target Recognition (ATR) application. ATR involves processing images for detecting, classifying, and tracking targets embedded in a background scene. We address the problem of discriminating between targets and non-target objects in a scene by evaluating 40x40 image blocks belonging to an image. Each image block is first projected onto a set of templates specifically designed to separate images of targets embedded in a typical background scene from those background images without targets. These filters are found using directed principal component analysis which maximally separates the two groups. The projected images are then clustered into one of n classes based on a minimum distance to a set of n cluster prototypes. These cluster prototypes have previously been identified using a modified clustering algorithm based on prior sensed data. Each projected image pattern is then fed into the associated cluster's trained neural network for classification.

A detailed description of our algorithm will be given in this paper. We outline our methodology for designing the templates, describe our modified clustering algorithm, and provide details on the neural network classifiers. Evaluation of the overall algorithm demonstrates that our detection rates approach 96% with a false positive rate of less than 0.03%.

1. Introduction

There has been much work involved in the process of automatic target recognition (ATR). This process involves automatic detection, classification, and tracking of a target located, or camouflaged, in an image scene. The typical procedure utilized for recognition involves a three-stage process - segmentation, feature extraction, and classification. The segmentation process is useful for dividing the image space into separate regions of interest. The feature extraction process allows the ATR system to identify and classify targets based on relevant

features and the classification process involves detecting and consequently identifying the target in question.

An ATR system must be invariant toward vantage points. This includes illumination changes, shadowing, perspective distortion, and occlusion. In Aerial ATR applications, the input image is typically an on-line aerial image acquired by digital camera. Such real world imagery is affected by climate, season, weather, and time of day. An aerial image is also subject to geometric changes, such as position, orientation, and scale variations. There are many other problems which face ATR systems. Normally, the target recognition process is highly data dependent. Most systems are only capable of recognizing a pre-specified number of targets and are unable to expand their object database. In addition, many ATR systems are encoded with predetermined tolerances resulting in a tendency to be very sensitive to scale and orientation changes. MODALS [11], a 3-D multiple object detection and location system, utilizes a neural network to simultaneously segment, detect, locate, and identify multiple targets. Although MODALS is able to provide robust detection, high classification, and a low false alarm rate, it is not rotation or scale invariant. SAHTIRN [1] performs automatic target recognition through a three-stage process using an edge detector, a multi-layer feedforward clustering neural network and a neural network classifier. SAHTIRN is able to successfully classify objects with varying scale and orientation parameters, but is not robust when faced with changes in lighting conditions. Greenberg and Guterman[5] use neural networks to address the issue of target classification, but assume the ATR-detection process has previously executed and has already identified targeted regions of interest.

Our research objective is to develop a novel technique which autonomously detects, in real time, all target objects embedded in a background image scene. The evaluation of these algorithms is based on inserting target images into real scenes acquired

from video input. In real time, we will reduce the data dimensionality of a scene using an optimal set of templates and spatially locate targets in the scene with a neural network classifier. Figure 1 provides an overview of our approach for detecting a known set of targets in a background image. The rest of this paper describes the methodology used to investigate autonomous target detection in detail.

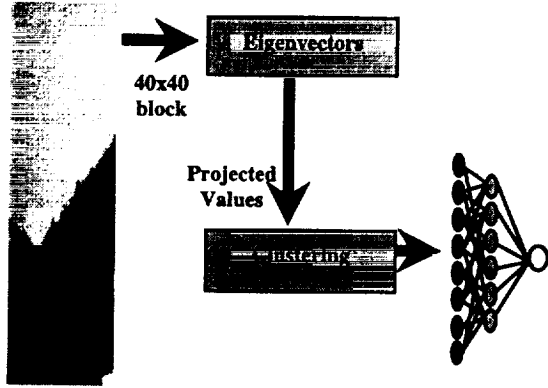


Figure 1: The data processing path for each 40x40 image block extracted from the acquired video input. The image block is projected onto a set of filters, associated with a particular cluster, and then classified with the associated neural network.

2. Technique

A. Background and Target Data Set

The background image scenes used in this research effort are acquired from video camera from the JPL in Pasadena site. We segment these background images into 40x40 image blocks for input into our algorithm (Fig. 2:Top). Target objects (Fig. 2:Middle) are modeled from an actual cruise missile and represent various scale and rotation perspectives of the missile. These synthetic target objects are used for training the algorithm and are embedded into the background image such that:

$$I^T = \begin{cases} b_{xy} : t_{xy} = 0 \\ 0.6t_{xy} + 0.4(t_{x-1,y} + t_{x+1,y} + t_{x,y-1} + t_{x,y+1}) : t_{xy} > 0 \end{cases}$$

where t_{xy} is the pixel intensity value of the target image at (x,y) , b_{xy} is the pixel value of the background image at (x,y) , and I^T is the embedded target-background image block. Example embedded

target-background images are shown in Figure 2:Bottom.

Once we extract our background and embedded target data set, we perform a preprocessing step in order to account for time-of-day lighting variations in the image set. We subtract the average image block intensity value from each pixel such that:

$$I^c = I - \frac{\sum i_{xy}}{N}$$

where i_{xy} is the image intensity of pixel (x,y) in image block I , N is the size of image block I (40x40), and I^c is the corrected image block to be used in our algorithm.

Using this data set, we can train an algorithm capable of intelligently detecting a target embedded within a background image scene. The next section describes our approach for the development of such an algorithm.

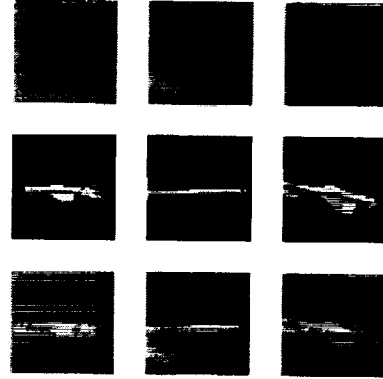


Figure 2. Top Row: Background Images; Middle Row: Target Objects; Bottom Row: Background Images with Embedded Targets

B. Algorithm Description

Given a set of targets T , the goal of the algorithm is to detect in real time, any target $t \in T$ present in a 40 x40 image block extracted from a background scene. After the data preprocessing step, we begin by projecting an image block onto a set of templates specifically designed to separate signatures derived from a target embedded in a background image from other typical background images. These projections, or patterns, are then clustered into one of n classes

based on their distance to a set of n cluster prototypes. These cluster prototypes have previously been identified using a modified clustering algorithm based on prior sensed data. Associated with each cluster is a trained neural network classifier. After clustering, the projected image pattern is fed through this associated trained neural network for detection.

In order to accomplish our target detection goal, prior knowledge must be derived through the following algorithmic preprocessing steps:

- i. Derive a set of linear filters used to optimally separate targets embedded in a background image from other background images.
- ii. Identify a set of cluster prototypes used to classify the projected image patterns.
- iii. Train a set of *expert* neural network classifiers for each cluster which responds with 1 when fed embedded target-background image patterns and -1 otherwise.

i. Linear Filter Sets

The filtering step involves an orthogonal sub-space projection of each image block. It is used to optimally linearly separate the embedded target background images from those images without targets. This is a standard technique used to reduce the dimensionality of the image block [from 1600 (40x40) to 17 dimensions] while preserving as much of the signal as possible. The filters associated with a given prototype are derived from the distribution of a background image (*noise*) and the distribution of potential targets embedded in that background (*signal*). This can be optimally separated to maximize the *signal to noise* ratio between the two groups using directed principal components analysis (DPCA). To characterize the distribution for the background image, the covariance matrix, R_i , is found for image blocks which do not contain targets. We characterize the mixed target-background image distribution instances by its covariance matrix, S_i .

We are interested in finding a set of orthogonal basis vectors W_i , that maximizes the expected signal to noise ratio of these two distributions defined by their respective image sets. The generalized eigenvector solution:

$$S_i W_i = \lambda R_i W_i$$

accomplishes this. The set of filters defined by W_i is the directed components used in our algorithm. They essentially steer the eigenvector solution away from dimensions of high noise variance in a linearly optimal fashion. Figure 3 shows a subset of the linear filter set.

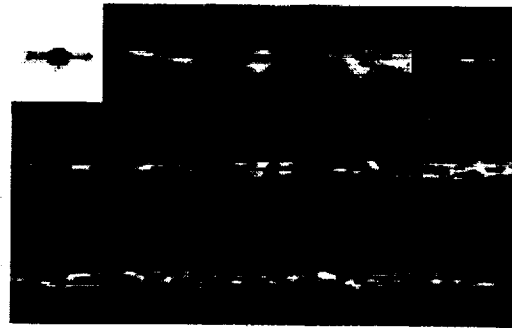


Figure 3. Top 15 directed principle components

ii. Clustering

To effectively simplify the distribution of data classified by an expert neural network, we partition the incoming projected image patterns drawn from a known distribution of background and embedded target-background images into a number of predetermined groups by using the prototypes P_i of a clustering algorithm. The clustering algorithm is run on previously acquired data that reflects the distribution of the scene being analyzed.

The clustering algorithm employed is a modified version of a standard clustering technique outlined in Duda and Hart [3]. The standard algorithm uses a standard least squares criterion to minimize the distance between each of n randomly selected groups. The criterion minimized by the standard clustering algorithm is:

$$(1) \text{ cost} = \sum_i \sum_j \| p_j - P_i \|^2$$

where i is one of n clusters and p_j is a projected image pattern in that cluster. The clustering algorithm iterates through each projected image pattern and determines if moving the pattern to another group reduces the overall cost. If it does, the pattern is moved to the other group and the associated averages of each prototype cluster are recalculated. This continues until the moving of patterns no longer reduces the overall cost. The

resultant cluster prototypes are then employed by our algorithm to segment the scene.

As n increases, the overall cost is likely to go down as a larger number of groups allow the clustering algorithm to better fit the given distribution. Secondly, the clustering algorithm, as described, is independent of the variation in the clustering set. It does not take into account any information that we might have concerning the patterns already belonging to the cluster. What we need is a way to penalize a cluster when adding a pattern that is different than the majority of patterns already in the cluster. This, in effect, will allow the clustering algorithm to more likely group embedded target-background patterns together while still discarding those background patterns which may have similar characteristics. To accomplish this, we modified the criterion given in (1) to reflect this knowledge. The change in (1) consists of simply weighting the initial criterion by a term reflecting variance. The modified criterion is given by:

$$(2) \text{ cost} = \sum_i (1+w_i) \sum_j \|p_j - P_i\|$$

where

$$w = \frac{nt_i}{np_i} \left(1 - \frac{nt_i}{np_i}\right)$$

nt_i is the number of embedded target patterns in cluster i , and np_i is the number of background patterns in cluster i . Patterns that are not like those already in the cluster will be weighed more in the cost of the clustering algorithm than those alike thus allowing the clustering algorithm to naturally link

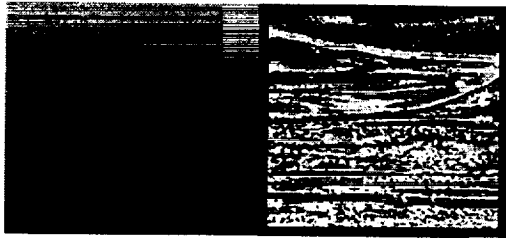


Figure 4. A Segmented Background Image Scene

alike elements together. Figure 4 shows a background scene segmented with the derived clustering prototypes.

iii. Classification

The last step in our preprocessing algorithm involves classifying each projected image pattern belonging to cluster i with a neural network [6]. The networks are trained with data drawn from the two distributions: background patterns R_i and embedded target-background patterns S_i . The expert network for class i is required to respond with 1 for elements drawn from S_i and -1 from those drawn from R_i . We use a simple feed forward network model employing 17 inputs and 10 sigmoidal hidden units trained with backpropagation to get the desired result. The output can then be thresholded to achieve the desired detection rate or false positive rate by examining the receiver operator curves.

C. Algorithm Implementation

After we implement the preprocessing steps, we can perform real-time intelligent target detection. After subtracting out the mean, each image block is projected onto a linear filter set. This projected image pattern is then compared to the set of pre-computed cluster prototypes. Based on the Euclidean distance, the pattern is grouped with the closest prototype.

We then use the trained neural network classifier to evaluate whether or not the image pattern contains a target from T . The neural network for each cluster group takes as input the projected values of the image and outputs a value. Values above a threshold are considered images with targets and those below are assigned to background.

The effectiveness of the evaluation requires that the cluster prototypes generated and the image blocks used in training the classifier must be derived from scenery with roughly the same distributions as encountered in the operational test. The following pseudo code outlines the important features of the algorithm.

Let I_{xy} be an image block with a centroid located at (x,y) in the scene:

for all x,y -

1. $I_{xy} \rightarrow f^c W$
2. $f^c W \rightarrow p_{xy}$
3. for $\min(\|p_{xy} - P_i\|)$
 4. if $N_i(p_{xy}) < \text{thr}_i$ then $\text{target}(x,y)$
 - otherwise $\text{background}(x,y)$

where f^c is the mean corrected image block, W is the linear filter set, p_{xy} is the projected image block, P_i is

the closest cluster prototype, N_i is the neural network classifier associated with cluster i , and thr_i is the threshold value discriminating between images with targets and those without.

3. Results

We evaluated the overall performance of the algorithm using the described target set and background images.

The background scenes consisted of over one million image blocks of which less than 5% were used in developing a set of training data. Testing data consisted of randomly drawn image blocks from the background scenes. Embedded target images were generated by randomly selecting images from the target set and mixing them with arbitrary background image blocks. A sub-sample of the training data (1000 examples each) was used to generate the covariance matrixes R and S . The generalized eigenvector solution W was then solved using Matlab. The training data was then projected onto the filter set and evaluated with the clustering technique to realize the cluster prototypes (P) used in step i of the algorithm.

Training data for the neural network was again drawn from the set of training image blocks. In addition, a portion of the training data for the network was used to halt training (a hold out set) as described in Haykin [6]. Training of the networks used 80,000 examples, $\frac{1}{2}$ target and $\frac{1}{2}$ background images. The hold out set consisted of 40,000 examples not trained upon. It is used to stop training in order to prevent over learning on the training data which tends to decrease generalization.

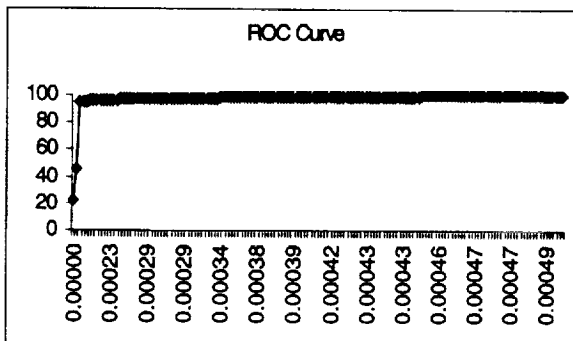


Figure 5: Detection vs False Positive Rate

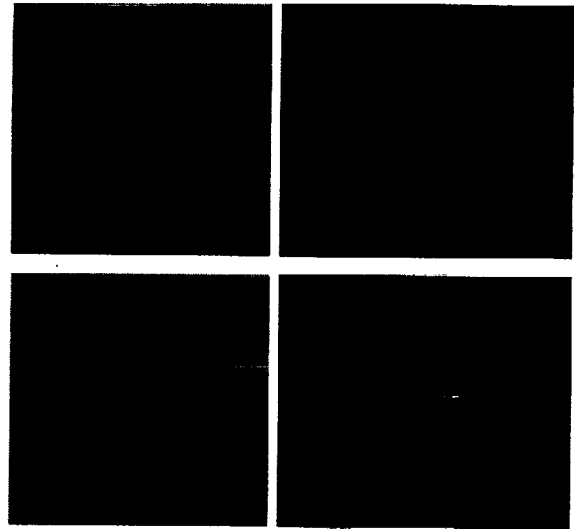


Figure 6: Detection Output

Our results give us a detection rate of 96% with a false positive rate of less than 0.03%. These results are constructed with 100,000 novel 40x40 image blocks. Figure 5 shows our Receiver Operator Curve (showing Detection vs. False Positive Rate) and Figure 6 shows an example of the detection output.

4. Conclusion

A novel detection algorithm and our evaluation methodology are described here. The detection algorithm was shown to perform detection at a rate of 96% with false positives less than 0.03% on a set of targets mixed with background images.

5. Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the Ballistic Missile Defense Organization through an agreement with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

- [1] C.E. Daniell, D.J. Kemsley W.P. Lincoln, W.A. Tackett, G.A. Baraghimian, "Artificial neural networks for automatic target recognition", *Optical Engineering*, Vol. 31, No. 12, pp. 2521-2530, Dec. 1992.
- [2] K. Diamantaras, S. Kung, *Principal Component Neural Networks: Theory and Applications*. John Wiley & Sons. New York. 1996.
- [3] H. Duda, M. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons. New York. 1973.
- [4] J.F. Gilmore, "Knowledge-based target recognition system evolution", *Optical Engineering*, Vol. 30, No. 5, pp. 557-570, May 1991.
- [5] S. Greenberg, J. Guterman, "Neural-network classifiers for automatic real world aerial image recognition", *Applied Optics*, Vol. 35, No. 23, pp. 4598-4608, Aug. 1996.
- [6] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company. New York. 1994.
- [7] A. Howard, C. Padgett, C.Liebe, "A Multi-Stage Neural Network for Automatic Target Detection" in proceedings 1998 IEEE World Congress on Computational Intelligence, May 98.
- [8] S.K. Rogers, J.M. Colombi, C.E. Martin, J.C. Gainey, K.H. Fielding, T.J. Burns, D.W. Ruck, M. Kabrisky, M. Oxley, "Neural Networks for Automatic Target Recognition", *Neural Networks*, Vol. 8, No. 7/8, pp. 1153-1184, 1995.
- [9] C. Padgett, G. Woodward, "A hierarchical, automated target recognition algorithm for a parallel analog processor", *Proc. of 1997 IEEE Int. Symposium on Computational Intelligence in Robotics and Automation '97*, Monterey Bay, CA, July 10-12.
- [10] C. Padgett, M. Zhu, S. Suddarth, "Detection and orientation classifier for the VIGILANTE image processing system", *Proc. of SPIE Int. Symposium on Aerospace/Defense Sensing and Controls*, Vol. 3077, April 1997.
- [11] W.A. Thoet, T.G. Rainey, D.W. Brettle, L.A. Slutz, F.S. Weingard, "ANVIL neural network program for three-dimensional automatic target recognition", *Optical Engineering*, Vol. 31, No. 12, pp. 2532-2539, Dec. 1992.
- [12] S. Udomkesmalee, A. Thakoor, C. Padgett, T. Daud, W-C Fang, S. Suddarth, "VIGILANTE: An Advanced Sensing /Processing Testbed for ATR Applications", *Proc. of SPIE Int. Symposium on Aerospace /Defense Sensing and Controls*, Vol. 3069, April 1997.



THE
OFFICE OF THE
ATTORNEY GENERAL
STATE OF NEW YORK
ALBANY, N. Y.
JANUARY 1, 1901